ISEN
ALL IS DIGITAL!
OUEST

Oxymètre de pouls

Projet Système Complexe S6

Partie informatique

Encadrement

Brest	Nantes	Caen
Pierre-Jean Bouvet	Florian Bescher	Simon Le Gloannec
François Legras	Nils Beaussé	
Thomas Paviet-Salomon		
Nabil Kadjouh		

Présentation

Description générale

Partie Electronique analogique

Module « INTERFACE CAPTEUR CARDIO »

LED-R

LED-IR

DET

LUM

Partie Electronique numérique

Module « MESURE ET AFFICHAGE CARDIO »

AC_R

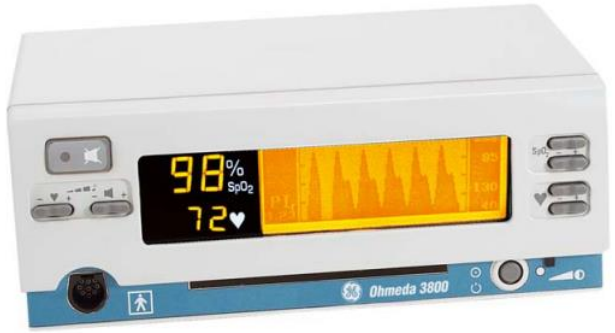
DC_R

AC_{IR}

DC_{IR}

Partie informatique

Module TRAITEMENT AFFICHAGE OXYMETRE

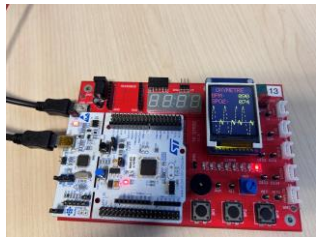


Description de la partie informatique

- Récupération des valeurs mesurées de la lumière rouge (ACR et DCR) et de l'infra-rouge (ACIR et DCIR) via liaison USB
- Traitement des informations reçues afin d'éliminer les signaux parasites
- Calcul de l'oxymétrie et de fréquence cardiaque
- Visualisation des résultats via une interface utilisateur

Synoptique de la partie informatique

Carte électronique (fournie)



USB

21o

Lecture

AC_R DC_R AC_{IR} DC_{IR}

FIR

AC_R DC_R AC_{IR} DC_{IR}

IIR

AC_R DC_R AC_{IR} DC_{IR}

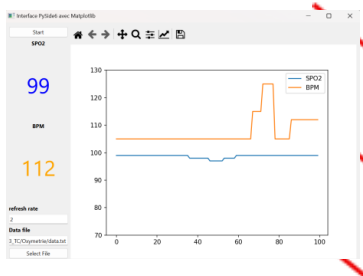
Mesure

SPO_2

Pouls

Affichage

Interface python (fournie)



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

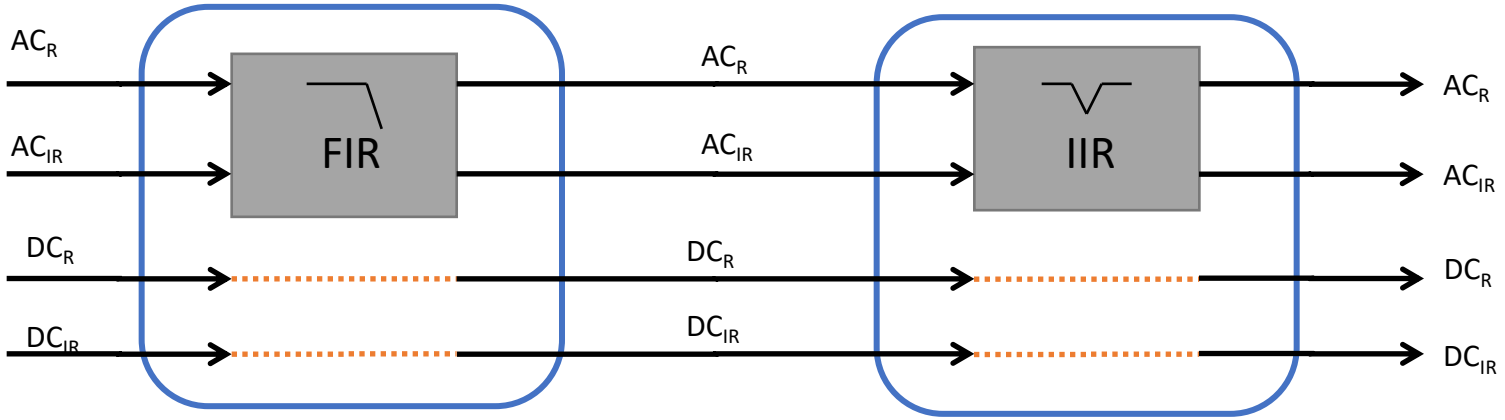
Fichiers test (fournis)

xx Module à développer pour tous

xx Module à développer pour les CIR3 (Bonus pour les autres)

Les différents blocs

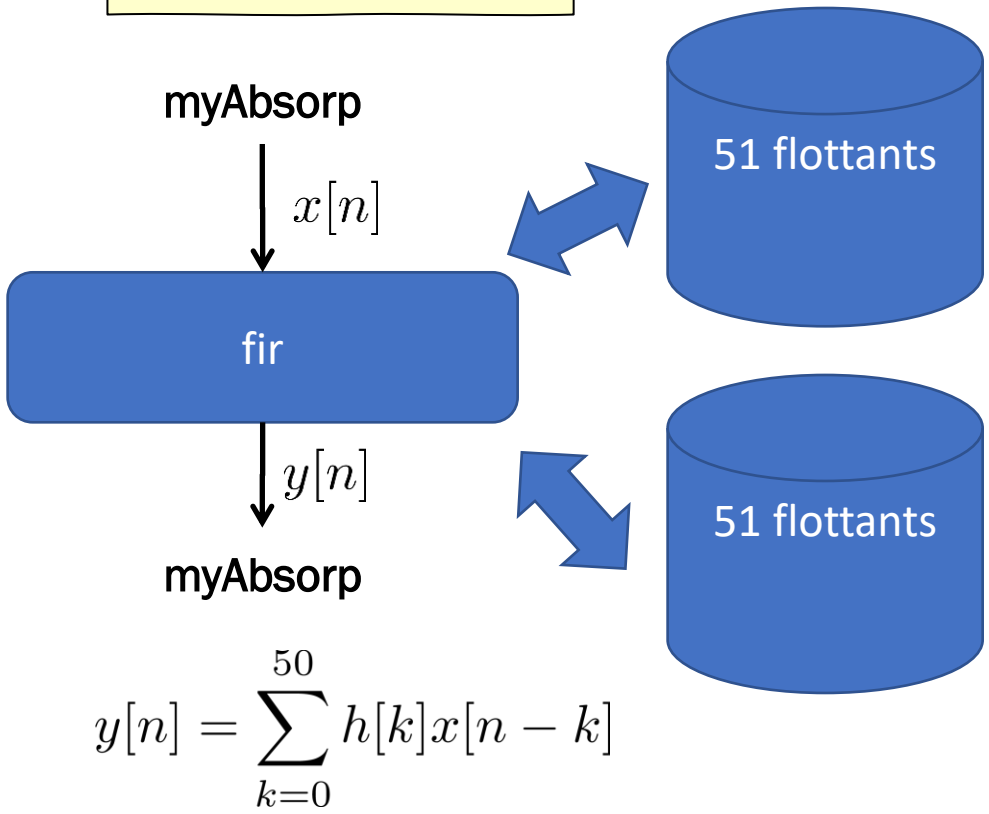
Filtrage (deux blocs : FIR et IIR)



- Points essentiels
  - Filtrage passe-bas (FIR) et passe-haut (IIR) des composantes AC_R et AC_{IR}
- Points critiques
  - Implémentation des filtres

Filtrage FIR

```
typedef struct{
          float acr;
          float dcr;
          float acir;
          float dcir;
} absorp;
```

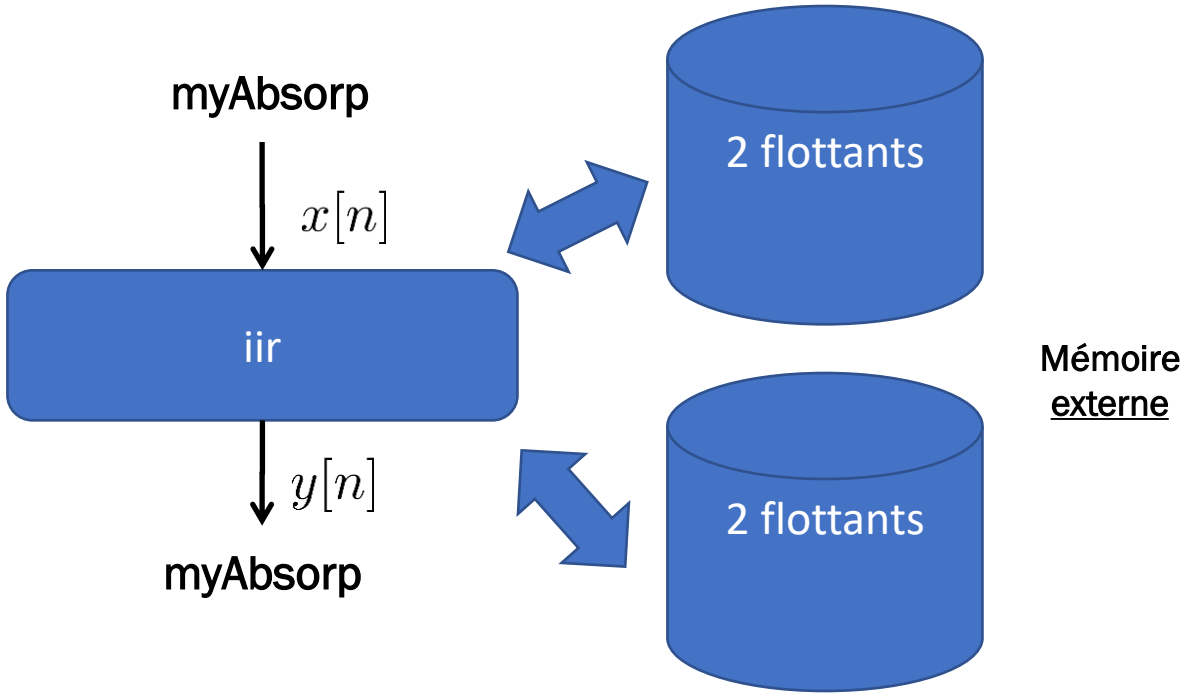


Mémoire externe

$$y[n] = \sum_{k=0}^{50} h[k]x[n-k]$$

k	$h[k]$	k	$h[k]$	k	$h[k]$
0	1.4774946e-019	17	3.1294938e-002	34	2.7892178e-002
1	1.6465231e-004	18	3.4578348e-002	35	2.4459630e-002
2	3.8503956e-004	19	3.7651889e-002	36	2.1082435e-002
3	7.0848037e-004	20	4.0427695e-002	37	1.7838135e-002
4	1.1840522e-003	21	4.2824111e-002	38	1.4793934e-002
5	1.8598621e-003	22	4.4769071e-002	39	1.2004510e-002
6	2.7802151e-003	23	4.6203098e-002	40	9.5104679e-003
7	3.9828263e-003	24	4.7081811e-002	41	7.3374938e-003
8	5.4962252e-003	25	4.7377805e-002	42	5.4962252e-003
9	7.3374938e-003	26	4.7081811e-002	43	3.9828263e-003
10	9.5104679e-003	27	4.6203098e-002	44	2.7802151e-003
11	1.2004510e-002	28	4.4769071e-002	45	1.8598621e-003
12	1.4793934e-002	29	4.2824111e-002	46	1.1840522e-003
13	1.7838135e-002	30	4.0427695e-002	47	7.0848037e-004
14	2.1082435e-002	31	3.7651889e-002	48	3.8503956e-004
15	2.4459630e-002	32	3.4578348e-002	49	1.6465231e-004
16	2.7892178e-002	33	3.1294938e-002	50	1.4774946e-019

Filtrage IIR

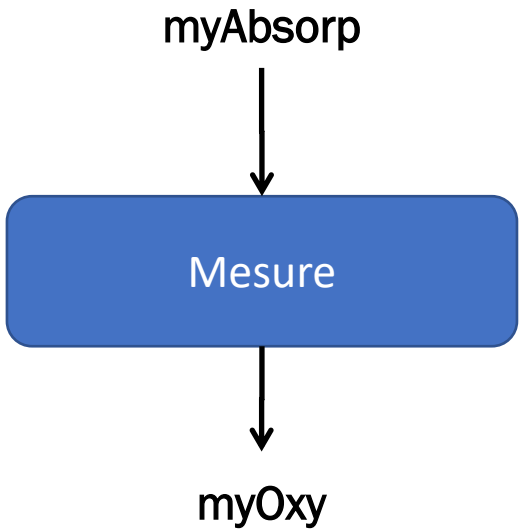


$$y[n] = x[n] - x[n-1] + \alpha y[n-1] \quad \text{avec} \quad \alpha = 0.992$$

Mesure

- Prend en entrée les valeurs mesurées de la lumière rouge (AC_R et DC_R) et de l'infra-rouge (AC_{IR} et DC_{IR})

- Extraction à partir des différentes composantes des quantités suivantes :
  - Taux de saturation d'oxygène dans le sang (SpO_2)
  - Fréquence cardiaque (pouls)

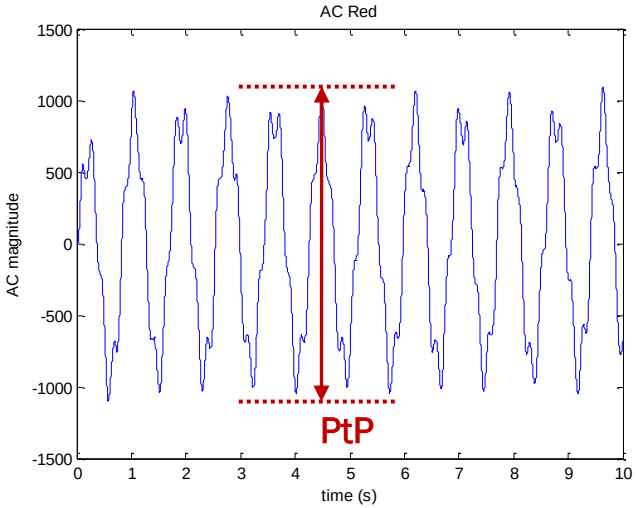


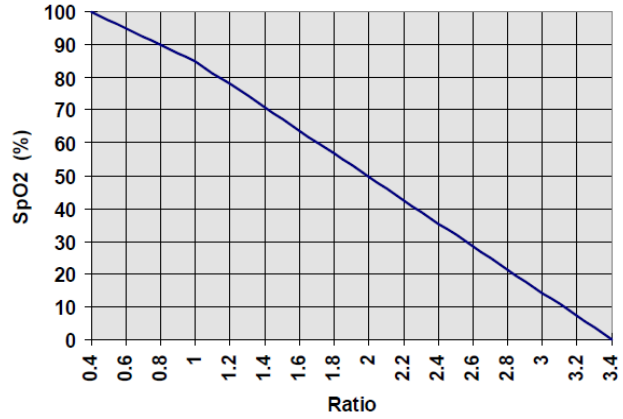

```
typedef struct{
          int spo2;   /*!< SPO2 */
          int pouls;   /*!< Pouls */
} oxy;
```

Mesure $Sp0_2$

- Points essentiels
  - Calcul de la valeur crête de chaque composante AC
  - Calcul de RsIR puis $Sp0_2$ via une table de correspondance
- Points critiques
  - Calcul de $Sp0_2$ au "fil de l'eau"

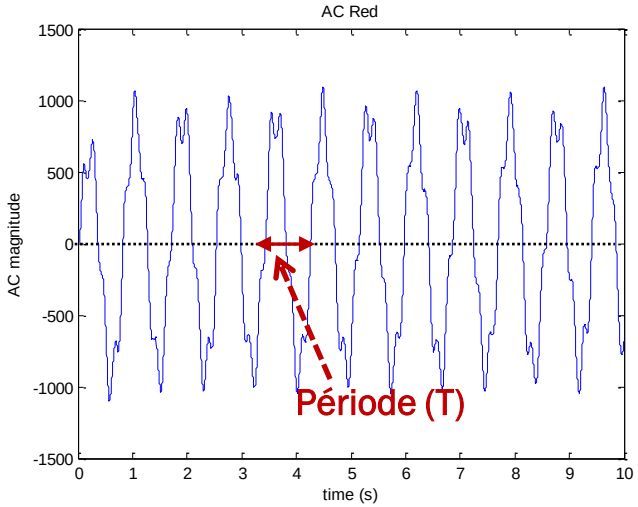
$$RsIR = \frac{\frac{\mathrm{PtP}(AC_R)}{DC_R}}{\frac{\mathrm{PtP}(AC_{IR})}{DC_{IR}}}$$





Mesure de la fréquence cardiaque

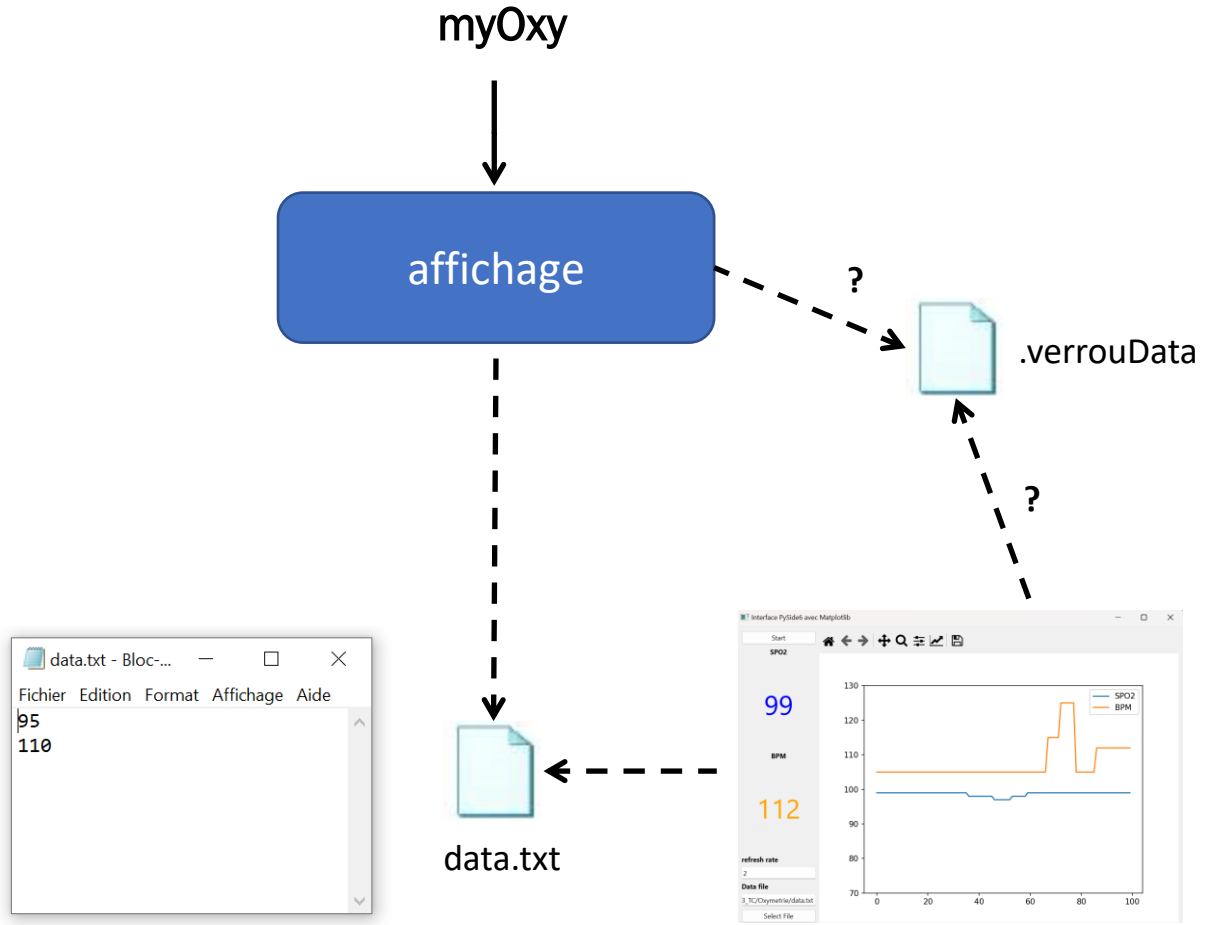
- Points essentiels
  - Calcul de la fréquence de l'onde de pouls (sur AC_R et AC_{IR})
- Points critiques
  - Calcul de la fréquence cardiaque au "fil de l'eau"



$$F = \frac{1}{T}$$

Affichage

- Points essentiels
  - Ecriture des informations dans le fichier data.txt
- Points critiques
  - Exclusion mutuelle à l'aide d'un verrou



Affichage

- Test existence d'un fichier

```
#include <unistd.h>

if( access( ".verrouData", F_OK ) != -1 )
{
        // Fichier existe
}else{
        // Fichier n'existe pas
}
```

- Effacer un fichier

```
remove(".verrouData")
```

Lecture

- Points essentiels
  - Lire chacun des champs
  - Recentrer les composantes AC autour de 0
- Points critiques
  - Etablir la synchronisation afin de récupérer les valeurs
  - Driver FTDI

Lecture

- Détail de la trame

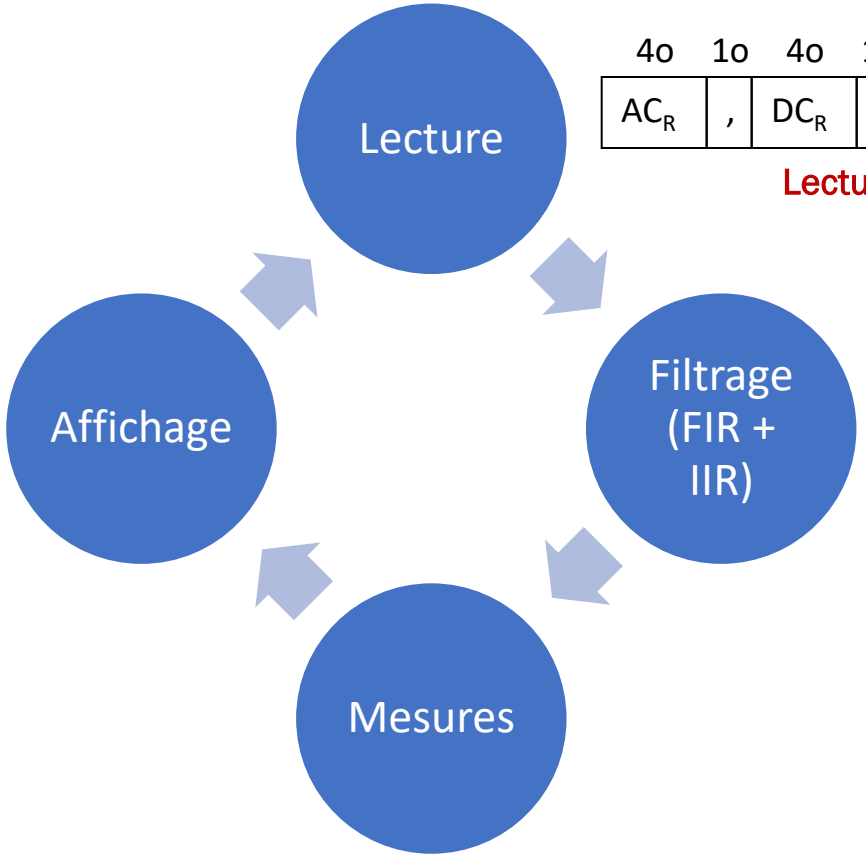
21 octets (rafraichissement toutes 2 ms)

Taille (octets)	4	1	4	1	4	1	4	1	1
Champ	AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Architecture

Organisation du programme

Le traitement se fait au fil de l'eau !



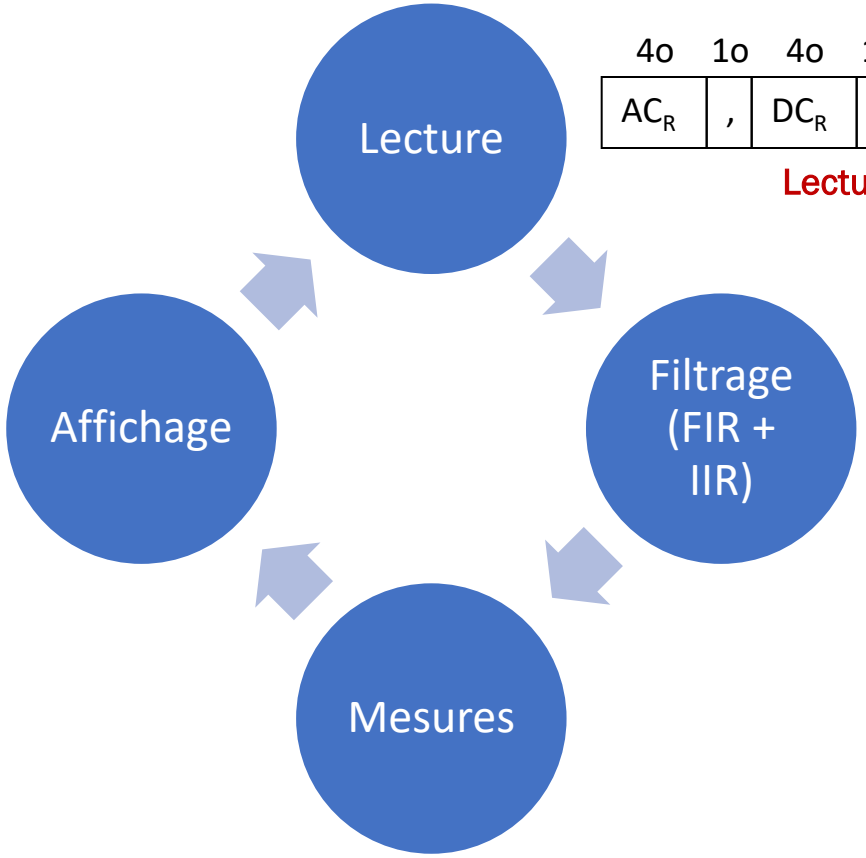
4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une mesure

$AC_R = 16$
$DC_R = 2030$
$AC_{IR} = 23$
$DC_{IR} = 2030$

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une mesure

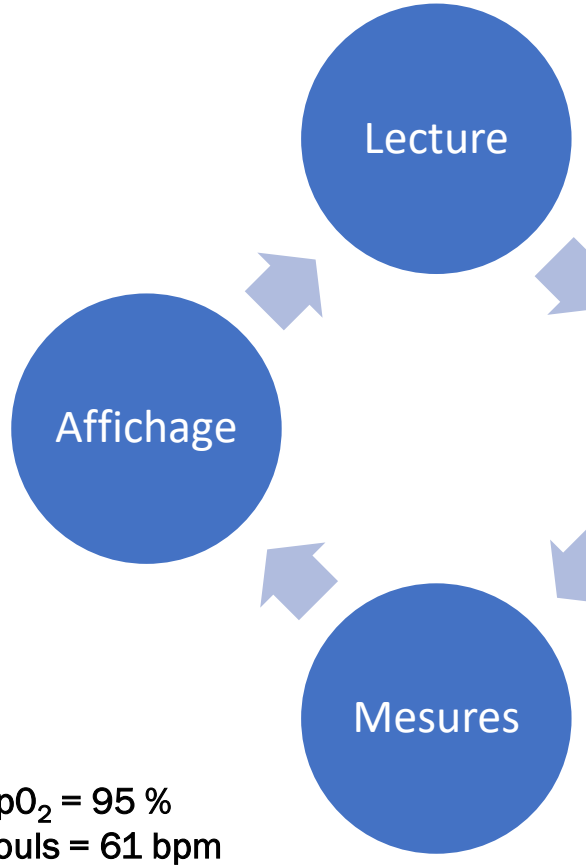
$AC_R = 16$
$DC_R = 2030$
$AC_{IR} = 23$
$DC_{IR} = 2030$

$AC_R = 10$
$DC_R = 2030$
$AC_{IR} = 30$
$DC_{IR} = 2030$

Filtrage d'une mesure

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une mesure

$AC_R = 16$
$DC_R = 2030$
$AC_{IR} = 23$
$DC_{IR} = 2030$

$AC_R = 10$
$DC_R = 2030$
$AC_{IR} = 30$
$DC_{IR} = 2030$

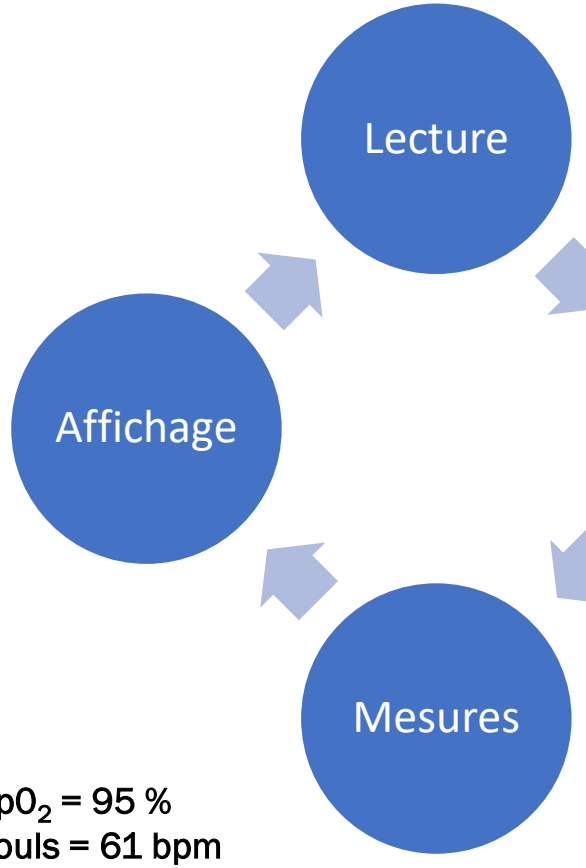
Filtrage d'une mesure

$Sp0_2 = 95\ \%$
Pouls = 61 bpm

Calcul de Sp02 et Pouls

Organisation du programme

Le traitement se fait au fil de l'eau !



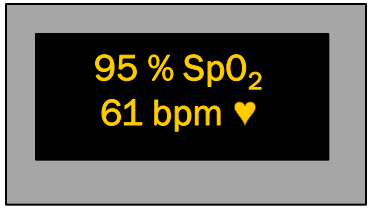
4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une mesure

$AC_R = 16$
$DC_R = 2030$
$AC_{IR} = 23$
$DC_{IR} = 2030$

$AC_R = 10$
$DC_R = 2030$
$AC_{IR} = 30$
$DC_{IR} = 2030$

Filtrage d'une mesure



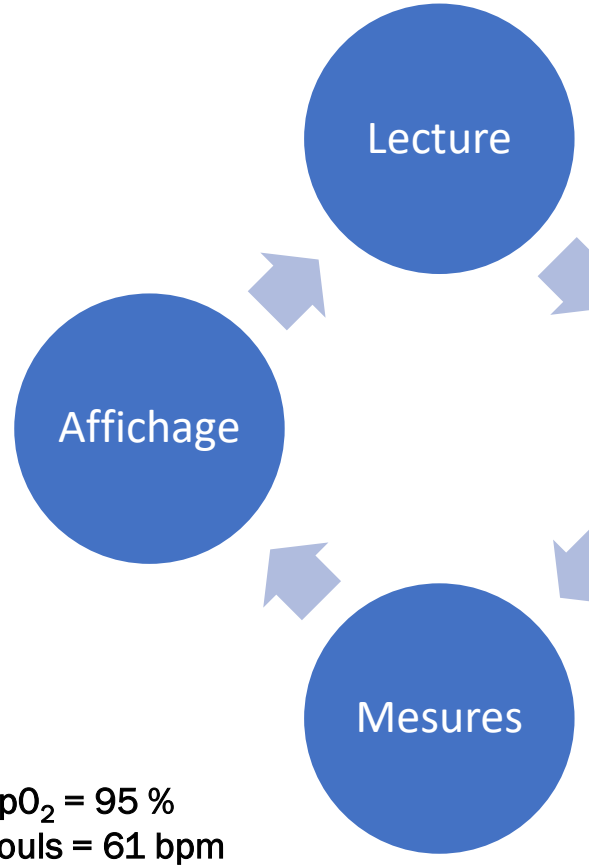
Affichage du résultat

$SpO_2 = 95$ %
Pouls = 61 bpm

Calcul de Sp02 et Pouls

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une nouvelle mesure

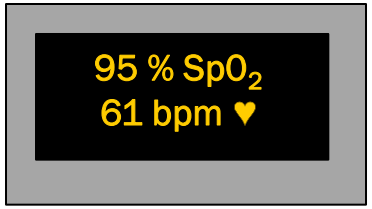
AC_R = 8
DC_R = 2029
AC_{IR} = 51
DC_{IR} = 2028

AC_R = 10
DC_R = 2030
AC_{IR} = 30
DC_{IR} = 2030

Filtrage d'une mesure

SpO_2 = 95 %
Pouls = 61 bpm

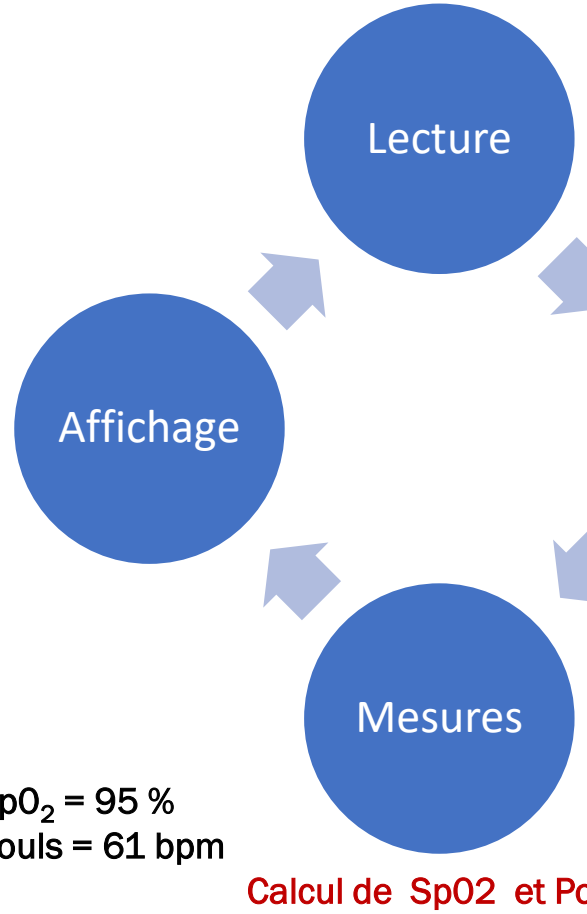
Calcul de Sp02 et Pouls



Affichage du résultat

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une nouvelle mesure

AC_R = 8
DC_R = 2029
AC_{IR} = 51
DC_{IR} = 2028

AC_R = 12
DC_R = 2029
AC_{IR} = 45
DC_{IR} = 2028

Filtrage d'une nouvelle mesure

SpO_2 = 95 %
Pouls = 61 bpm

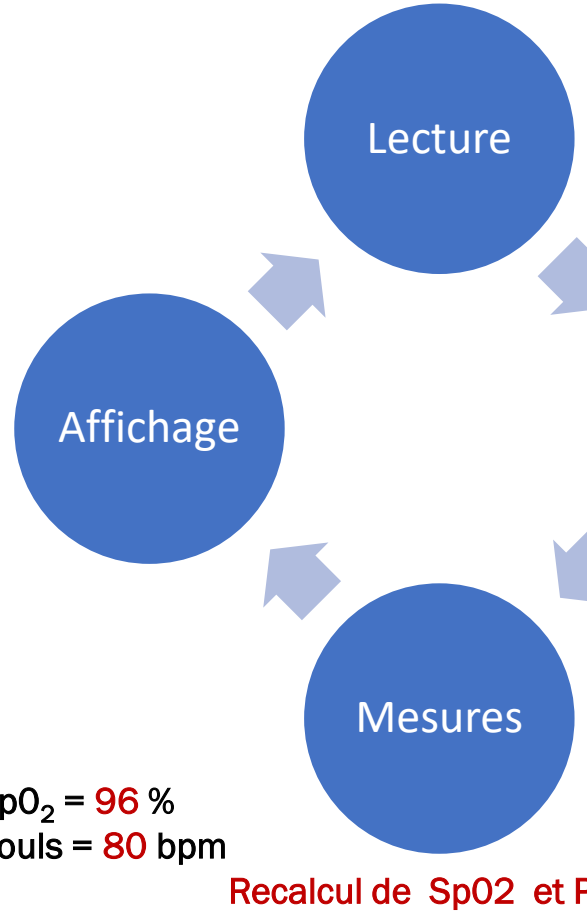
Calcul de Sp02 et Pouls

95 % SpO_2
61 bpm ♥

Affichage du résultat

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une nouvelle mesure

AC_R = 8
DC_R = 2029
AC_{IR} = 51
DC_{IR} = 2028

AC_R = 12
DC_R = 2029
AC_{IR} = 45
DC_{IR} = 2028

Filtrage d'une nouvelle mesure

SpO_2 = 96 %
Pouls = 80 bpm

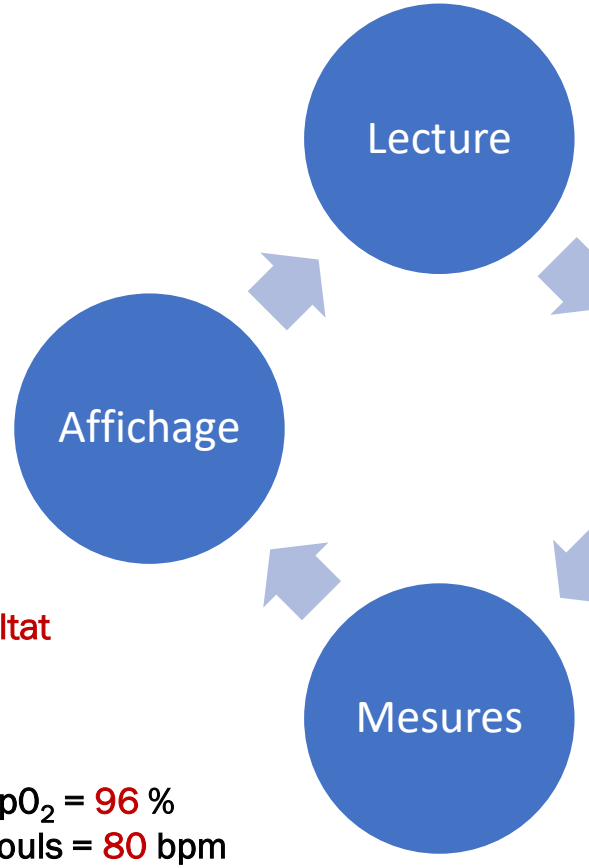
Recalcul de Sp02 et Pouls

95 % SpO_2
61 bpm ♥

Affichage du résultat

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une nouvelle mesure

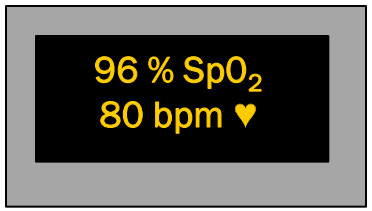
AC_R = 8
DC_R = 2029
AC_{IR} = 51
DC_{IR} = 2028

AC_R = 12
DC_R = 2029
AC_{IR} = 45
DC_{IR} = 2028

Filtrage d'une nouvelle mesure

SpO_2 = 96 %
Pouls = 80 bpm

Recalcul de Sp02 et Pouls

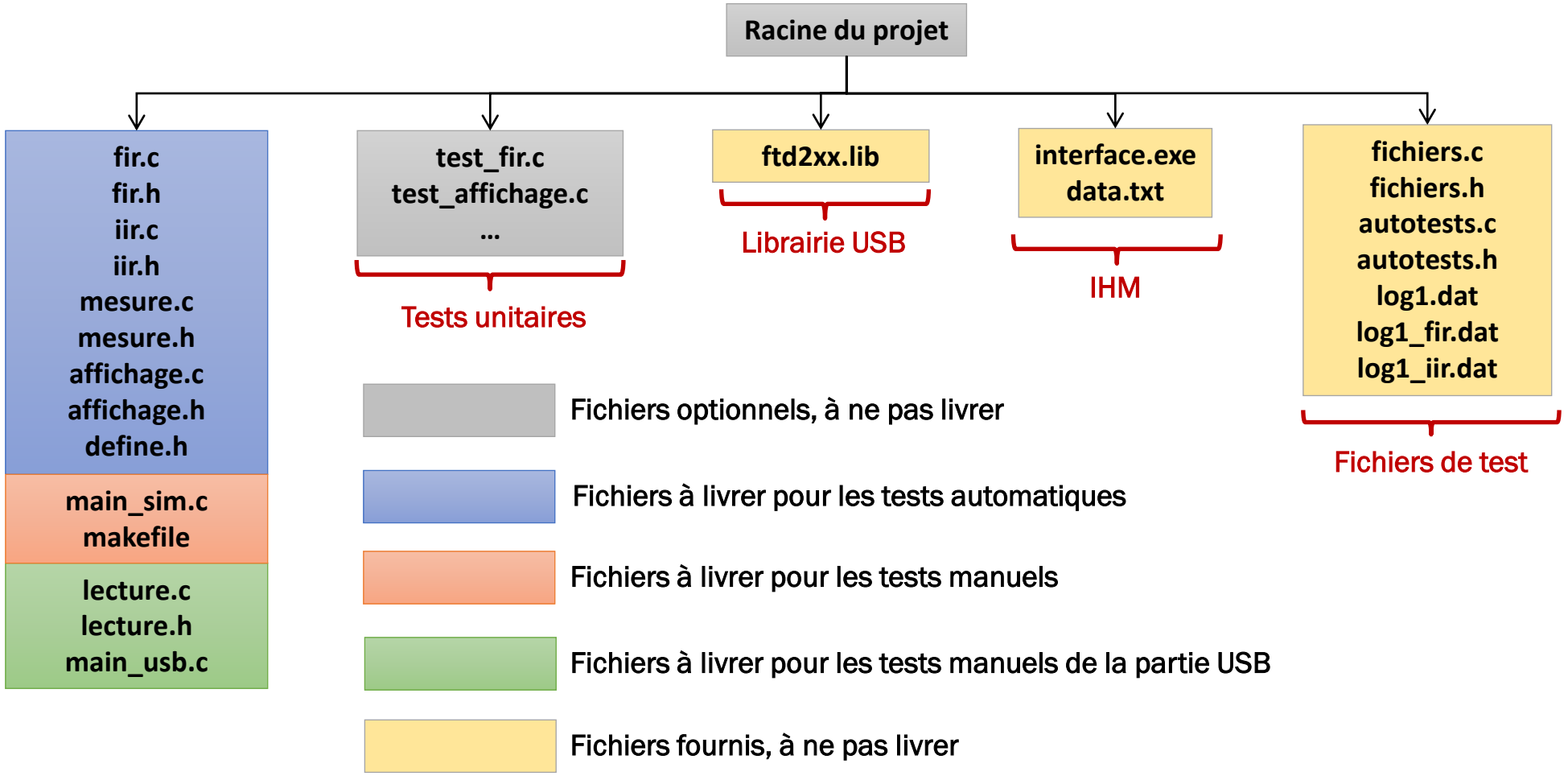


Affichage du nouveau résultat

Convention de nommage

Bloc	Nom fonction	Nom fichier source	Nom fichier header
Filtrage FIR	`fir`	`fir.c`	`fir.h`
Filtrage IIR	`iir`	`iir.c`	`iir.h`
Mesure	`mesure`	`mesure.c`	`mesure.h`
Affichage	`affichage`	`affichage.c`	`affichage.h`
Lecture	`lecture`	`lecture.c`	`lecture.h`
Définition globales			`define.h`

Arborescence



Exemple de programme

main.c

```
int main(){
        int etat=0;
        absorp myAbsorp;
        oxy myOxy;
        param_fir* myFIR = init_fir(…); // init FIR
        param_iir* myIIR = init_iir(…); // init IIR
        param_mesure* myMes = init_mesure(…) // init mesure
        FILE* myFile = initFichier(« log1.dat");
        do{
                myAbsorp = lireFichier(myFile,&etat);
                myAbsorp = fir(myAbsorp,myFIR);
                myAbsorp = iir(myAbsorp,myIIR);
                myOxy = mesure(myAbsorp,myMes);
                affichage(myOxy);
        }while( etat != EOF );
        finFichier(myFile);
        fin_mesure(myMes);
        fin_iir(myIIR);
        fin_fir(myFIR) ;
        return EXIT_SUCCESS;
}
```

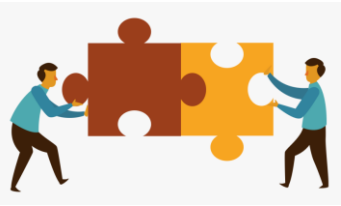
define.h

```
#ifndef DEFINE_H
#define DEFINE_H

typedef struct{
          float acr;  /*!< AC R */
          float dcr;  /*!< DC R */
          float acir;  /*!< AC IR */
          float dcir;  /*!< DC IR */
} absorp;
typedef struct{
          int spo2;  /*!< SPO2 */
          int pouls;  /*!< Pouls */
} oxy;

#endif
```

Déroulé du projet

Déroulé du projet

4 phases



Lundi matin	Lundi après-midi	Mardi matin	Mardi après-midi	Mercredi matin	Mercredi après-midi	Vendredi matin	Vendredi après-midi
Analyse	Réalisation				Intégration		Recette

✓Analyse de chaque bloc
✓Architecture du programme
✓Définition des Types de données

✓Code source de chaque bloc
✓Tests unitaires
✓Makefile

✓Code source programme principal
✓Test intégration avec le simulateur ou avec la carte
✓Validation avec l'interface

✓**Code source à rendre avant 14:00**
✓QCM de 14:00 à 14:30
✓Validation automatique et manuelle par l'enseignant sur PC enseignant à partir de 14:30
✓Audit

Au jour le jour

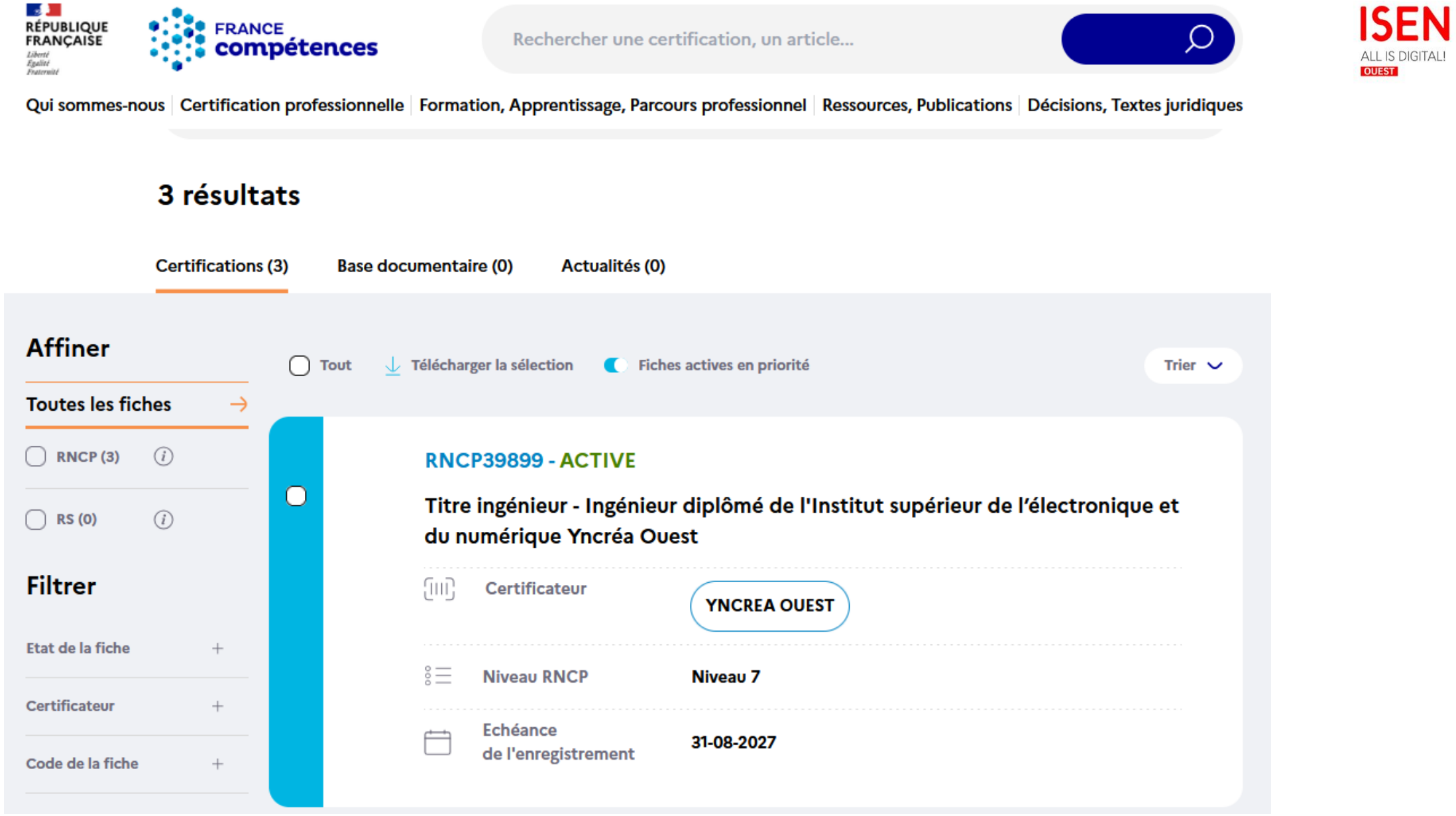
- Travail en binôme
  - Les binômes sont constitués le premier jour
    - Les binômes à constituer au sein de chaque sous-groupe
    - Les redoublants doivent se mettre ensemble ou en monôme le cas échéant
    - Pas 2 utilisateurs de macintosh au sein d'un même binôme
  - Chaque étudiant connait l'ensemble du projet
  - Attention à bien se répartir le travail
- Ressources externes
  - Tous les documents sont autorisés…
    - … mais les informations nécessaires au projet sont fournies !!!
  - La transmission de code entre étudiants de binômes différentes est interdite
    - un outil anti-plagiat sera utilisé en fin projet
  - L'utilisation d'outils de génération de codes par intelligence artificielle de type ChatGPT est totalement proscrit et sera sévèrement sanctionné, vous serez également noté sur votre compréhension du code.
- Livraison de code ou de document
  - Ne pas attendre la dernière minute pour poster le livrable
    - Préparer des livrables intermédiaires
    - Sauvegarder régulièrement vos données

ISEN
ALL IS DIGITAL!
OUEST

Evaluation

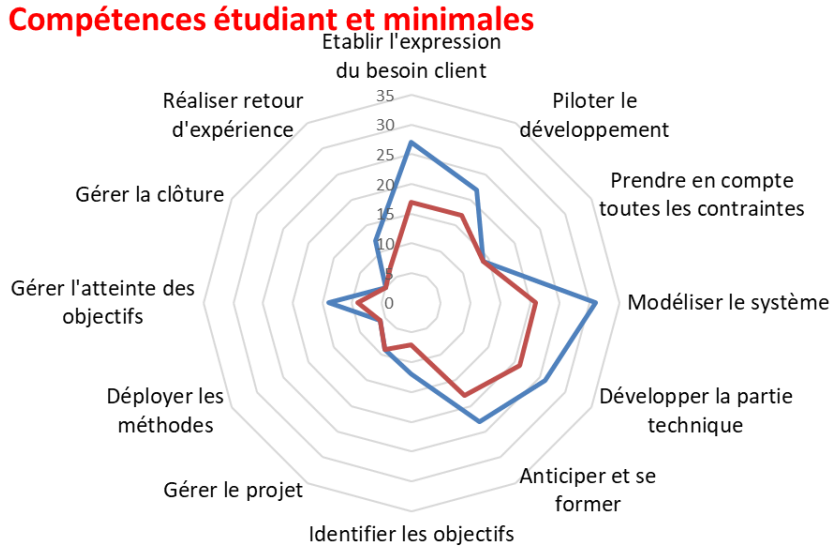
Pourquoi l'approche par compétences ?

- Montrer aux étudiants que leur formation sert à quelque chose
- Mettre en évidence les compétences techniques et professionnelles acquises en projet A3
- Mettre en évidence, sensibiliser et réfléchir au portfolio de compétences de chaque étudiant
- Améliorer le contenu technique (parag. Expérience pro, et compétences) du CV des étudiants
- Aider les étudiants dans leur recherche de stage/emploi


RÉPUBLIQUE FRANÇAISE
FRANCE compétences
Rechercher une certification, un article...
ISEN
ALL IS DIGITAL!
OUEST
Qui sommes-nous | Certification professionnelle | Formation, Apprentissage, Parcours professionnel | Ressources, Publications | Décisions, Textes juridiques
3 résultats
Certifications (3)
Base documentaire (0)
Actualités (0)
Affiner
Toutes les fiches
RNCP (3)
RS (0)
Filtrer
Etat de la fiche
Certificateur
Code de la fiche
Tout
Télécharger la sélection
Fiches actives en priorité
Trier
RNCP39899 - ACTIVE
Titre ingénieur - Ingénieur diplômé de l'Institut supérieur de l'électronique et du numérique Yncréa Ouest
Certificateur
YNCREA OUEST
Niveau RNCP
Niveau 7
Echéance de l'enregistrement
31-08-2027

Lien entre les cours/projets (organisés en UE/ECUE) et le référentiel de compétences ?

RNCP13040BC01 Définir les exigences et l'architecture du système électronique ou informatique			RNCP13040BC02 Réaliser le système électronique ou informatique			RNCP13040BC03 Piloter des hommes et des projets de systèmes électronique ou informatique					
LC1	LC2	LC3	LC1	LC2	LC3	LC1	LC2	LC3	LC4	LC5	LC6
Etablir l'expression du besoin client	Piloter le développement	Prendre en compte toutes les contraintes	Modéliser le système	Développer la partie technique	Anticiper et se former	Identifier les objectifs	Gérer le projet	Déployer les méthodes	Gérer l'atteinte des objectifs	Gérer la clôture	Réaliser retour d'expérience



Le tableau croisé !

Tableau croisé des compétences

	RNCP13040BC01 Définir les exigences et l'architecture du système électronique ou informatique			RNCP13040BC02 Réaliser le système électronique ou informatique			RNCP13040BC03 Piloter des hommes et des projets de systèmes électronique ou informatique					
	LC1	LC2	LC3	LC1	LC2	LC3	LC1	LC2	LC3	LC4	LC5	LC6
	Etablir l'expression du besoin client	Piloter le développement	Prendre en compte toutes les contraintes	Modéliser le système	Développer la partie technique	Anticiper et se former	Identifier les objectifs	Gérer le projet	Déployer les méthodes	Gérer l'atteinte des objectifs	Gérer la clôture	Réaliser retour d'expérience
L'étudiant·e sait developper des fonctions en respectant un cahier des charges		x	x		x		x	x	x	x		
L'étudiant·e sait intégrer différentes fonctions		x			x			x	x	x		
L'étudiant·e maitrise son sujet sur les aspects techniques.				x		x						
L'étudiant·e maitrise le langage de programmation					x	x						
L'étudiant·e est capable de prendre du recul sur le sujet du projet			x	x		x						
L'étudiant·e est capable d'expliquer l'entrée/la sortie (le type et le sens) d'une fonction				x	x						x	x
L'étudiant·e est capable d'expliquer, de manière globale, le fonctionnement global d'une fonction (son déroulé, son utilité globale)				x	x						x	x
L'étudiant·e est capable d'expliquer une fonction ligne par ligne				x	x						x	x
L'étudiant·e se soucie de la suite de son projet (documentation…)									x		x	x
L'étudiant·e sait respecter des consignes									x		x	x

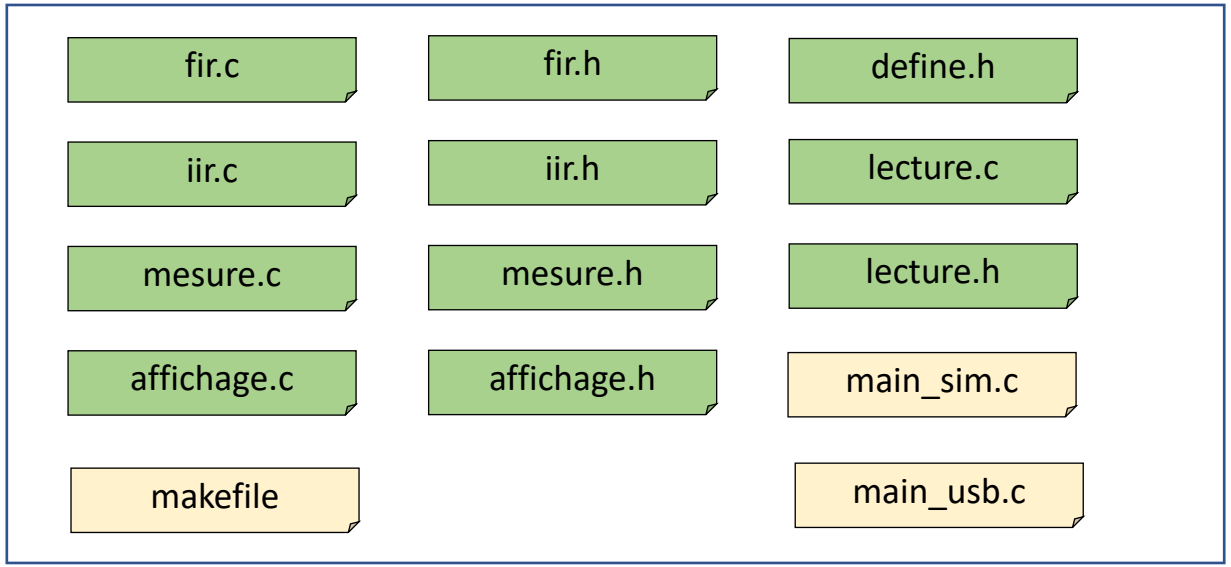
Evaluation par compétences

Méthode d'évaluation	Compétences évaluées	Principe de validation	Pondération
Recette	L'étudiant sait développer des fonctions en respectant un cahier des charges	tests automatiques Blocs FIR, IIR, Affichage et calcul	4,5
Recette	L'étudiant sait intégrer intégrer différentes fonctions	Tests manuels en simulation et USB	3
QCM papier	L'étudiant maitrise son sujet sur les aspects techniques.	Note de QCM	2
QCM papier	L'étudiant maitrise le langage de programmation	Note de QCM	2
QCM papier	L'étudiant est capable de prendre du recul sur le sujet du projet	Note de QCM	2
Audit	L'étudiant·e est capable d'expliquer l'entrée/la sortie (le type et le sens) d'une fonction	Questions aux étudiants	1,5
Audit	L'étudiant·e est capable d'expliquer, de manière globale, le fonctionnement global d'une fonction (son déroulé, son utilité globale)	Questions aux étudiants	1,5
Audit	L'étudiant·e est capable d'expliquer une fonction ligne par ligne	Questions aux étudiants	1,5
Audit	L'étudiant se soucie de la suite de son projet (documentation…)	Audit de code	1
Audit	L'étudiant sait respecter des consignes	Audit de fichiers (Noms des fonctions, nommage des fichiers, makefile, …)	1
Total			**20**

- **Pour la recette, aucune validation au regard du code source**
- **Seulement sont testées les fonctionnalités d'exécution de votre code**

Ce qu'il faut rendre

- Livrable



- Chaque fichier doit être commenté
- Le nommage des fonctions doit être respecté

Le code source

- Chaque bloc :
  - Un fichier
  - Une fonction

Les interfaces de la fonction principale de chaque bloc sont imposées afin de permettre une validation automatique lors de la recette. Les prototypes vous seront fournis en début de réalisation !

- Programme principal
  - Permet de tester chaque bloc de façon indépendante (ne doit pas être remis)
  - Permet de tester le programme complet (obligatoire pour tests manuels)
    - Avec le simulateur
    - Avec la carte électronique
- Makefile (obligatoire seulement pour tests manuels)
  - Permet de compiler automatiquement votre code source
- Audit
  - Lors de la recette, il vous sera demandé d'expliquer une ou plusieurs portions de votre code

Les ressources

Le matériel

- Votre PC portable
  - Analyse
  - Réalisation
  - Intégration

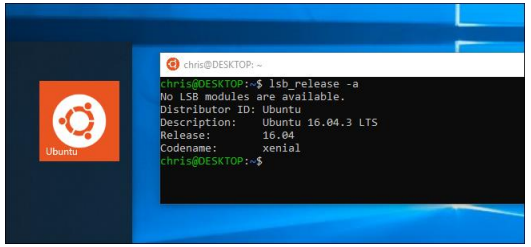
Casque / écouteur audio déconseillés (consignes non appliqués = pénalité) !

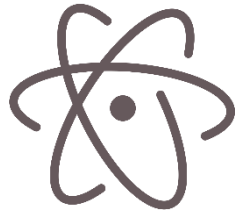
Boissons et nourriture interdites !

- Carte électronique
  - Validation de la partie USB

Outils de développement

- Noyau de compilation
  - Windows Subsystem Linux
  - Jupyterhub
  - Linux natif
- Editeur
  - Notepad++
  - Sublime Text
  - Atom
  - Visual studio code
  - …





Tests automatiques

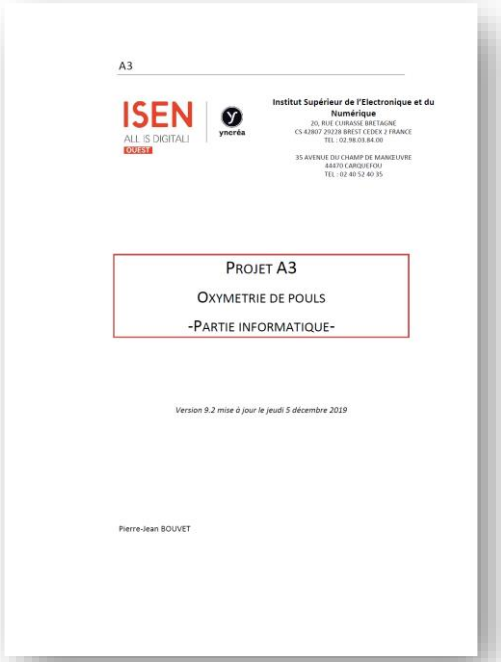
- Nous vous mettons à disposition un fichier de fonctions (autotests.c/autotests.h) permettant de tester unitairement les différents blocs :
  - testFIR()
  - testIIR()
  - testMesure()
  - testAffichage()

```
---- Autotest results of block FIR ----
--> test 0 : 1/1
--> test 1 : 1/1
--> test 2 : 1/1
--> test 3 : 1/1
Finish autotest of block FIR => total score : 100.0 %
---- Autotest results of block IIR ----
--> test 0 : 1/1
--> test 1 : 1/1
--> test 2 : 1/1
--> test 3 : 1/1
Finish autotest of block IIR => total score : 100.0 %
---- Autotest results of block Mesure ----
--> test 0 : 1/1
--> test 1 : 1/1
Finish autotest of block Mesure => total score : 100.0 %
warning : .verrouData exists, data could not be written
---- Autotest results of block Affichage ----
--> test 0 : 1/1
--> test 1 : 1/1
Finish autotest of block Affichage => total score : 100.0 %
```

Documentation

- Support de présentation

- Description détaillée



- Guide de programmation FTDI

ISEN
ALL IS DIGITAL!

yncréa

MERCI

Des questions ?